(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平8－279758

(43)公開日　平成８年(1996)10月22日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｈ０３Ｍ	7/30		9382－5K	Ｈ０３Ｍ	7/30	Ｂ
Ｇ１０Ｌ	7/04			Ｇ１０Ｌ	7/04	Ｇ
	9/18				9/18	Ｈ

審査請求　未請求　請求項の数１　ＦＤ　（全 14 頁）

(21)出願番号　特願平7－107957

(22)出願日　平成７年(1995)４月７日

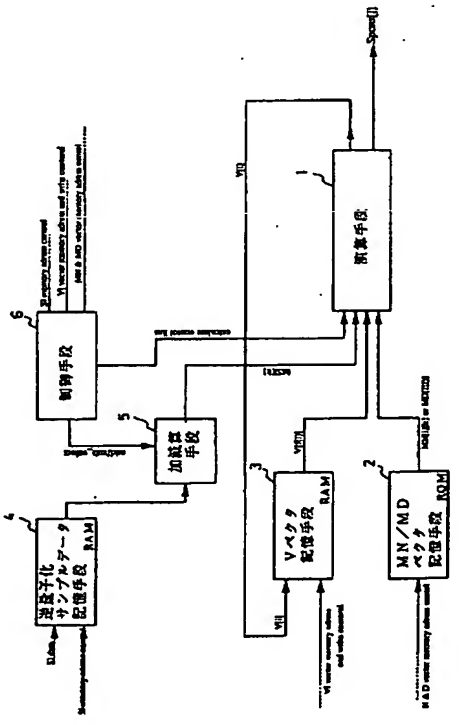
(71)出願人　000000952
鐘紡株式会社
東京都墨田区墨田五丁目17番４号

(72)発明者　岡本　彰
大阪市都島区友渕町１丁目６番７－304号

(72)発明者　下山　雅通
大阪市都島区友渕町２丁目12番21－303号

(54)【発明の名称】　音声信号帯域合成復号化装置

(57)【要約】
【目的】ＭＰＥＧ規格を用いた圧縮音声／オーディオ信号の伸張において、ハードウェア規模の縮小、高速化を図ることこと。
【構成】サブバンド生成用のＶベクタを算出する演算の冗長的な部分を削除して保持に必要な領域を縮小し、かつ係数の変換を行って、各演算手段に共通性を持たせ、同一のハードウェアで時分割処理を行う。



1

【特許請求の範囲】

【請求項１】　ＭＰＥＧ規格を用いた圧縮音声信号の伸長装置であって、音声データＰＣＭを算出する際に必要となる係数であるＮベクタ及びＤベクタに符号変換を行ったＭＤベクタを予め格納したＮ／ＭＤベクタ記憶手段、逆量子化データを加算もしくは減算して修正逆量子化データを算出する加減算手段、修正逆量子化データ及びＮベクタを用いたサブバンド合成用のＶベクタの算出と、ＭＤベクタ及びＶベクタを用いた音声データＰＣＭの算出とを同一演算手法を用いて時分割処理により行う演算手段、Ｖベクタを記憶するＶベクタ記憶手段と、これら各手段を制御する制御手段とからなることを特徴とする音声信号帯域合成複号化装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明はＭＰＥＧ規格の高能率符号化方式を用いた圧縮音声／オーディオの伸張装置に関し、更に詳しくは、比較的小規模なハードウエアで高速に伸張処理できる装置に関する。

【０００２】

【従来の技術】最近では、マルチメディアへの使用を目的とした動画像／音声の高能率符号化が必要不可欠となっている。この高能率符号化の国際標準規格としてＭＰＥＧ（Ｍｏｖｉｎｇ　Ｐｉｃｔｕｒｅ　Ｅｘｐｅｒｔｓ　Ｇｒｏｕｐ）方式があり、既にＩＳＯ／ＩＥＣ　１１１７２として勧告され、ＭＰＥＧ規格を用いた圧縮音声／オーディオ信号の伸張方法に関しては、「ＦＡＳＴ　Ｓｕｂｂａｎｄ　Ｆｉｌｔｅｒｉｎｇ　ｉｎ　ＭＰＥＧ　Ａｕｄｉｏ　Ｃｏｄｉｎｇ（ＩＥＥＥ　ＳＩＧＮＡＬ　ＰＲＯＣＥＳＳＩＮＧ　ＬＥＴＴＥＲ，Ｖｏｌ　１，ＮＯ　２，ＦＥＢＲＵＡＲＹ　１９９４）」等にも記載されている。

【０００３】かかるＭＰＥＧ方式による圧縮音声／オーディオ伸張に際しての帯域合成処理の原理手順を説明する。図１０はＩＳＯ／ＩＥＣ　１１１７２の勧告に示されているＭＰＥＧ方式のオーディオレイヤー２の規格にしたがった帯域合成のフローチャートである。ＭＰＥＧ方式では３２のサンプルデータごとに逆量子化が行われ、帯域合成処理が開始される（ステップ１０１）。次のステップ１０３では生成される６４個の新たなＶベクタに備えて、１０２４個からなるＶベクタ群中、古い６４個がシフト処理され（ステップ１０２）、入力された３２の逆量子化サンプルデータから新たなＶベクタが６４個算出される（ステップ１０３）。次に１０２４個のＶベクトル群から５１２個のＵベクタ群が生成され（ステップ１０４）、Ｕベクタ群にウインドウ処理に用いられる係数である係数Ｄｉが乗じられて、ウインド処理されてＷベクタ群が生成される（ステップ１０５）。Ｗベクタ群は、所定方向に１６個加算されて１サンプル分の音声／オーディオデータ（３２ワード）が伸張され（ステップ１０６）、これが３６回繰り返して実行されて３６グループからなる１フレームが伸張される。

2

【０００４】また、特開平６－７７８３９号公報等には、かかる演算やハードウェア規模を縮小するために、逆量子化ステップを量子化情報が共通な１２グループ単位で演算を行い、１フレーム単位で一括して帯域合成処理を行う方法が提案されている。

【０００５】かかる圧縮伸長を、前記した勧告の規格通りにハードウエアに適用した場合以下のような問題点が生ずる。音声データＰＣＭサンプルの合成に際して、Ｖベクタ群、Ｕベクタ群、Ｗベクタ群を生成する必要があるため、処理手順が多く、使用するメモリも多くなる。各演算過程に共通性がないので回路の共有ができず、各演算に応じたハードウエアが必要となり、装置全体としてみればハードウエア規模が大きく、高価なものとなり民製品への利用の障害となる。

【０００６】また、特開平６－７７８３９号公報では、１フレーム単位で一括処理を行うことによってハードウェアの縮小を図っているが、逆量子化データ、Ｖベクタとして３２６４ワード分を必要とし、さらに１フレームごとの処理のため処理遅延が大きいという欠点が存在する。

【０００７】本発明は前記課題を解決するため、サブバンド生成用のＶベクタを算出する演算の冗長的な部分を削除して保持に必要な領域を縮小し、かつ係数（Ｄベクタ群）の変換を行って、各演算手順に共通性を持たせることによって、ハードウェア規模の縮小を図ることを目的とする。

【０００８】

【課題を解決するための手段】本発明は、ＭＰＥＧ規格を用いた圧縮音声信号の伸長装置であって、音声データＰＣＭを算出する際に必要となる係数であるＮベクタ及びＤベクタに符号変換を行ったＭＤベクタを予め格納したＮ／ＭＤベクタ記憶手段、逆量子化データを加算もしくは減算して修正逆量子化データを算出する加減算手段、修正逆量子化データ及びＮベクタを用いたサブバンド合成用のＶベクタの算出と、ＭＤベクタ及びＶベクタを用いた音声データＰＣＭの算出とを同一演算手法を用いて時分割処理により行う演算手段、Ｖベクタを記憶するＶベクタ記憶手段と、これら各手段を制御する制御手段とからなることを特徴とする音声信号帯域合成複号化装置である。

【０００９】

【作用】本装置では、逆量子化データを修正して用いると共に音声データＰＣＭを算出する際に必要とされるＤベクタを変換して用いることにより、サブバンド合成用のＶベクタの算出手法と音声データＰＣＭの演算手法を同一のものとし、両手段を共用する。

【００１０】即ち、Ｖベクタの算出は後述するＮベクタの変換と逆量子化データＳの修正を行うことにより次式

で求められる。

【0011】

【数1】

$0 <= i <= 31$

$$V512[i] = \sum_{k=0}^{15} (MN[i][K] * MSi(K))$$

【0012】一方、Dベクタを変換したMDベクタを用いると、音声データPCM（Sj）の算出は次式により求めることができる。

【0013】

【数2】

$0 <= i <= 31$

$$Sj[j] = \sum_{i=0}^{15} [V512(F1(i, j) * MD(F2(i, j)))]$$

【0014】両式は、乗算とシグマ加算とからなるものであるため、同じ演算手法（演算回路）を共用することができる。

【0015】

【実施例】以下、本発明の帯域合成複合化装置について詳述する。図1は本装置の基本的な装置構成を示すブロック図である。同図に示す如く、本装置は、サブバンド合成用のVベクタの算出手段と音声データPCMの演算手段を共用する演算手段1を中心として、該演算手段にデータを入力する手段として、MN／MDベクタ記憶手段2、演算手段から出力されるVベクタを格納するVベクタ記憶手段3、逆量子化サンプルデータを格納した逆量子化サンプルデータメモリ4からの出力を加減算して修正逆量子化サンプルデータを出力する加減算手段5を有し、これら各手段のコントロールを行う制御手段6とからなる。

【0016】以下、各手段について、順次詳述する。図2は、逆量子化サンプルデータメモリ4の内部構成を示すブロック図である。このブロックは逆量子化されたサンプルデータを保持するブロックで、1グループ分（36グループで1フレームを構成）を保持する32ワードのメモリ41（RAM）と、メモリのリード／ライトを制御するコントローラー42が存在し、出力するデータのアドレスは制御手段により制御される。

【0017】加減算手段5は、Vベクタを算出する前処理として、逆量子化されたデータを加算／減算するものであり、図3は、同手段の内部構成を示すブロック図である。逆量子化サンプルデータは、後述する式に基づいて特定アドレスの逆量子化サンプルデータと加算又は減算が施されるが、加減算手段には、1ワードづつ入力を行うのが装置規模を縮小するためには好ましく、このため加算／減算器51の一方にはレジスタ52が設けられ、制御装置からの指示によって、連続して入力された2つの逆量子化サンプルデータが加算／減算処理される。

【0018】Vベクタ記憶手段3は、後述する演算装置によって算出されたVベクタを保持するメモリ31で、512ワードのメモリに、新規作成される32個のVベクタを含む512個のVベクタを保持する。Vベクタ記憶手段の内部構成を図4に示す。同手段は、メモリを初期設定（0番地から480番地までデータ値をゼロに設定）するブロック32と、Vベクタのアドレス番地を書き換える（シフティング）ブロック33、メモリのライト／リード動作を制御するメモリコントロールブロック34で構成されており、メモリに使用するアドレス値や初期設定命令、シフティング命令は制御手段から制御される。

【0019】MN／MDベクタ記憶手段2は、乗算に使用する係数であり勧告で定められたNベクタ及びDベクタについて、これを変換したMNベクタ及びMDベクタを予め格納するもので、本実施例では、ROMを用いている。MN／MDベクタ記憶手段の内部構成を図5に示す。

【0020】MNベクタは後述するNベクタの冗長的な係数部分を削除したものであり、MDベクタはDベクタに、後述する符号変換関数を用いて符号変換を行ったものであり、さらに、Vベクタと同様のアドレス関数を使用することができるようにアドレスの番地の変換も施したものであり、該アドレスは制御装置により制御される。

【0021】制御手段6は、本装置の各種制御である初期設定やシフティング、積和演算、メモリのアドレスや動作制御等を統括する手段であり、その内部構成を図6に示す。同図に示す如く、本実施例では、外部から帯域合成スタート命令を受けて、マスタカウンタ61を稼働させ、カウンター値をデコードして、逆量子化サンプルデータメモリ制御デコーダ62、加減算手段制御デコーダ63、Vベクタ記憶手段制御デコーダ64、N／MDベクタ記憶手段制御デコーダ65、演算手段制御デコーダ66の夫々を起動する信号を発生する。

【0022】各デコーダはまずVベクタ記憶手段デコーダを起動し、後述する図11ステップ201のシフティング動作を行う。次に、逆量子化サンプルメモリ制御デコーダ62、加減算手段制御デコーダ63、Vベクタ記憶手段デコーダ64、MN／MDベクタ65の記憶手段デコーダ及び演算手段制御デコーダ66を起動し、図11ステップ202のVベクタ算出／記憶を行う。次にVベクタ記憶手段デコーダ64、MN／MDベクタの記憶

手段デコーダ６５及び演算手段制御デコーダ６６を起動し、図１１ステップ２０３の出力ＰＣＭ信号を算出する。

【００２３】演算手段１は、Ｖベクタの算出及びＰＣＭデータの算出に使用する積和演算器であり、その内部構成を図７に示す。本発明では、後述する係数等を採用することにより２つの積和演算は同一の演算手法、演算量となるため、演算手段では、時分割処理を行い同一回路を用いて演算を行う。よって入力値を選択するセレクタ１２ａ，１２ｂを設けて修正逆量子化サンプルデータとＭＮベクタの組み合わせと、ＶベクタとＭＤベクタの組み合わせを選択する。また、積和演算器１１の制御やセレクトライン等は制御手段により制御されている。

【００２４】次に、本装置で用いる帯域合成のアルゴリズムについて説明する。かかるアルゴリズムは、ＩＳＯ／ＩＥＣ　１１１７２の勧告に示されているＭＰＥＧ方式のオーディオレイヤー２の規格にしたがった帯域合成の演算手法を基本として、次の４項目について改良を加えることを特徴としている。以下、各改良項目について詳述する。

【００２５】（１）Ｖベクタの算出

図１０の勧告に示される処理フローでは、まず前回までのＶベクタ群のアドレス位置を変更する処理（ｓｈｉｆｔｉｎｇ）が行われた後、新規のＶベクタ（６４個）が算出される。勧告に示されるＶベクタの算出は以下の式の如くである。

【００２６】

【数３】

$0<=i<=15、33<=i<=47$

$k=2n$　のとき＋

$k=2n+1$のとき－

$$V[i]=\sum_{k=0}^{15}(N[i][k]*(S[K]\pm S[31-k]))$$

$i=16$

$$V[16]=0$$

$17<=i<=32$

$$V[i]=-V[32-i]$$

$i=48$

$$V[i]=\sum_{K=0}^{15}(S[K]+S[31-k])$$

$49<=i<=63$

$$V[i]=V[96-i]$$

【００３２】結果、必要とする演算量は、４９６回（１６×３１）の乗算と４８０回（１５×３２）の和、及び前処理としてそれぞれ１６回の和・差、及び後処理としての符号変換が１６回となり、演算に必要なＮベクタは図８に示された３２行×１６列のマトリクスとなり、これをＭＮベクタと定義する。

$0<=i<=63$

$$V[i]=\sum_{k=0}^{31}(N[i][K]*S[K])$$

【００２７】また、この式に用いられる係数Ｎ［ｉ］［ｋ］は以下の式によって定義されている。

【００２８】

【数４】

$0<=i<=63$　　$0<=k<=31$

$$N[i][k]=\cos[(16+i)(2k+1)\pi/64]$$

【００２９】かかる式から分かるように新規のＶベクタを算出するには、２０４８回（３２×６４）の乗算と１９８４回（３１×６４）の和が必要となってくる。ところが、この係数Ｎ［ｉ］［ｋ］を［ｉ＝行］［ｋ＝列］としたマトリクス形式で表現すると以下の様な性質があることが解る。

【００３０】１７行から３２行までの係数は、０行から１５行までの係数と対象であり、逆符号の関係である。４９行から６３行までの係数は、３３行から４７行までの係数と対象の関係でる。１６行目の係数はは常に「零（０）」であり、４８行目の係数は「－１」である。１６列から３１列までの係数は、０列から１５列までの係数と奇数行が対象、偶数行が逆符号対象の関係である。以上の性質を利用し、Ｖベクタの算出は図８に示すように変形され、次式の様に変形される。

【００３１】

【数５】

【００３３】（２）Ｖベクタ群の履歴の保持

Ｖベクタは、勧告では新規に作成された６４個のＶベクタと過去１５回分のＶベクタ（６４×１５＝９８０）の合わせて１０２４個のＶベクタを保持する必要がある。

7

しかし、前述の如くＶベクタは、符号だけが違い、絶対値は同じであるデータが約半数を占めているため、新規に作成されたＶベクタを例に取ると、図８に示されるようにデータ内容の重複あるいは符号が反転しているものが存在することが確認でき、保持すべきＶベクタは６４個の内、半分の３２個で良いことが解る。よって全体では保持すべきＶベクタ群は１０２４個（６４×１６）の内、５１２個（３２×１６）となり、シフトの回数も９＊

8

＊６０回から４８０回となり、処理時間、保持領域メモリを半減せしめることが可能となる。前述の性質より、１０２４個保持するＶベクタ群をＶ１０２４、５１２個保持するＶベクタ群をＶ５１２とすると、２つのベクタ群の関係は次の式の如くとなる。

【００３４】

【数６】

0<=k<=63とすると

V1024[64*N+k]=V512[32*N+k]　　(0<=k<=15)
=0　　(k=16)
=V512[32*N+(32-k)]　　(17<=k<=32)
=V512[32*N+(k-17)]　　(33<=k<=48)
=V512[32*N+(79-k)]　　(49<=k<=63)

0<=k<=31とすると

V1024[64*N+k]=V512[32*N+k]　　(0<=k<=15)
=0　　(k=16)
=V512[32*N+(32-k)]　　(17<=k<=32)
V1024[64*N+32+k]=V512[32*N+k]　　(k=0)
=V512[32*N+(k+15)]　　(1<=k<=16)
=V512[32*N+(47-k)]　　(17<=k<=31)

【００３５】（３）中間ベクタ（Ｕベクタ・Ｗベクタ）作成の削除

勧告によるフローでは、作成されたＶベクタから中間的なベクタ群（Ｕベクタ・Ｗベクタ）を抽出・算出して音声データＰＣＭを算出する。勧告による音声データＰＣＭを算出する式を以下に示す。

【００３６】

【数７】

0<=j<=31

$$Sj[j]=\sum_{j=0}^{15}W(j+32*i)$$

0<=i<=7
0<=j<=31
U[i*64+j]=V1024[i*128+j]
U[i*64+32+j]=V1024[i*128+96+j]

0<=i<=511
W[i]=U[i]*[i]

【００３７】この式に対して、以下の式を用いて変形し、それぞれのベクタのアドレスを示す関数（Ｆ１、Ｆ２、Ｆ３）を定義する。

【００３８】

【数８】

【００３９】この結果、前記式は以下の様に変換される。

【００４０】

【数９】

$$0 <= j <= 31 \qquad Sj[j] = \sum_{i=0}^{15} [F3(i, j) * V512(F1(i, j)) * D(F2(i, j))]$$

F1，F2：アドレス関数

F3　　：符号変換関数

【００４１】また、この変形過程を図９に示す。かかる式に定義したアドレス関数を用いればＶ５１２ベクタ群からU／Wベクタに相当するデータを読み出すことができるので、U／Wといった中間ベクタ群を生成する手順を省略できる。

【００４２】（４）Ｄベクタ郡の改良

前記式を用いて音声データＰＣＭを算出する場合、Ｖ５１２ベクタから抽出したベクタに符号変換関数（Ｆ３関数）を用いて符号変換を行った後、Ｄベクタと乗算する。符号変換関数およびＤベクタは、Ｖ５１２ベクタと＊同一アドレスで一対一に対応する性質を有するので、符号変換関数とＤベクタを乗算したものを保持するだけで、Ｖベクタの符号変換演算を削除できる。よって符号変換関数とＤベクタを乗算した結果を、新係数ＭＤベクタ（ｍｏｄｉｆｉｅｄ　Ｄｖｅｃｔｏｒ）と定義すると、以下の式の様になり、Ｖベクタを算出したときと同様の積和演算で済むこととなる。

【００４３】

【数１０】

$$0 <= j <= 31 \qquad Sj[j] = \sum_{i=0}^{15} [V512(F1(i, j)) * MD(F2(i, j))]$$

【００４４】図１１は、以上の改良を採り入れた本装置で用いる帯域合成のアルゴリズムを示す処理フローチャートである。先ず、逆量子化されたデータが１グループ（３２ワード）分入力される（ステップ２０１）。而して、以下のステップでは、前記（１）、（２）の項で説明したデータの性質を用いて、特開平６－７７８３９号公報がフレーム単位で処理を行っていたのに対し、３２個のデータを単位として処理を行う。即ち、ステップ２０２では生成される３２個の新たなＶベクタに備えて、５１２個からなるＶベクタ群中、古い３２個がシフト処理され、入力された逆量子化サンプルデータから新たなＶベクタが３２個算出される（ステップ２０３）。従って、新たに保持すべきＶベクタは５１２個で良い。

【００４５】次に前記（３）（４）の項で説明したアルゴリズムを用いて、Uベクタ群、Wベクタ群の生成を省略して１グループ分の音声／オーディオデータ（３２ＰＣＭサンプルデータ）が伸張され（ステップ２０３）、これが３６回繰り返して実行されて１フレームが伸張される。

【００４６】以上の如きアルゴリズムでは、ＭＰＥＧ方式規格を準拠しつつ冗長的な演算を削除したため、ハードウェア化した場合、以下のような特徴を有する。係数Ｎの冗長性を利用した積和演算を行うために演算量が半減し高速処理が可能となる。保持するデータ（Ｖベクタ）の冗長的な部分を保持しておく必要がないため、保持領域が半減し、メモリアクセス量を減少させることができる。音声データを生成する際に必要となる中間ベクタ（U／Wベクタ）を生成する必要がないため処理フローを簡素化することできる。Ｖベクタの算出方式における冗長部分の削除およびＤベクタを符号変換した新係数の採用により、音声データを作成する演算がＶベクタを作成する過程と同様の積和演算（乗算とシグマ加算）となるため、演算器を共有することが可能になり、ハードウエアの構成を簡素化することができる。

【００４７】

【発明の効果】本発明では、帯域合成復号化装置の冗長的な演算処理を削除したアルゴリズムを採用することにより、デコード期間の短縮／高速化と、ハードウエア規模の縮小化が図れ、同等の機能を安価に製造することができ、本装置は、ＭＰＥＧ規格による圧縮音声の伸張を民生用途に利用する際に頗る有用なものである。

【図面の簡単な説明】

【図１】本装置全体の基本的な構成を示すブロック図である。

【図２】本装置の逆量子化サンプルデータメモリの内部構成を示すブロック図である

【図３】本装置の加減算手段の内部構成を示すブロック図である。

【図４】本装置のＶベクタ記憶手段の内部構成を示すブロック図である。

【図５】本装置のＮ／ＭＤベクタ記憶手段の内部構成を示すブロック図である。

【図６】本装置の制御手段の内部構成を示すブロック図である。

【図７】本装置の演算手段の内部構成を示すブロック図である。

【図８】Ｖベクタの算出過程を示す説明図である。

【図９】Ｖベクタの算出過程を示す説明図である。

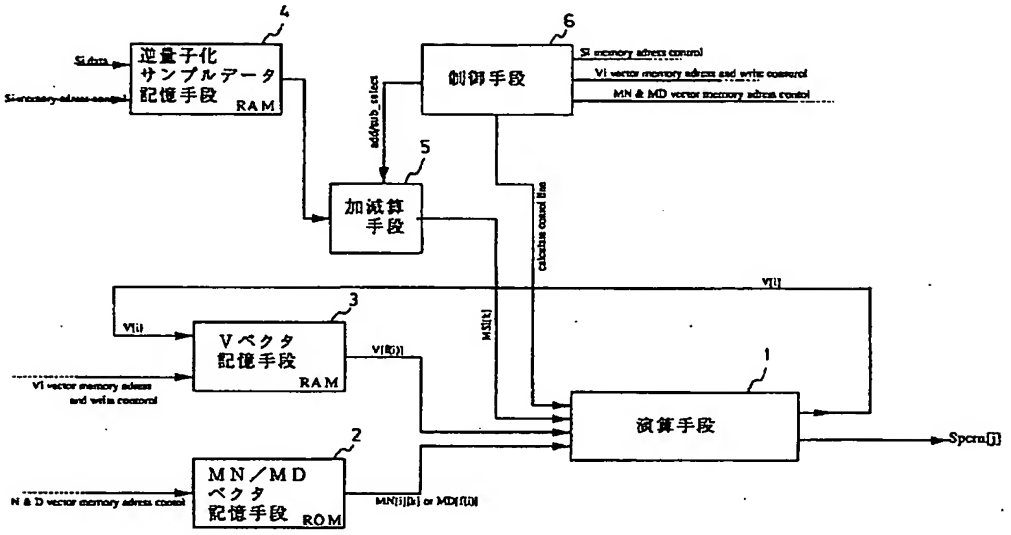
11

【図１０】ＭＰＥＧ方式の勧告に示されている帯域合成処理手順を示すフローチャートである。

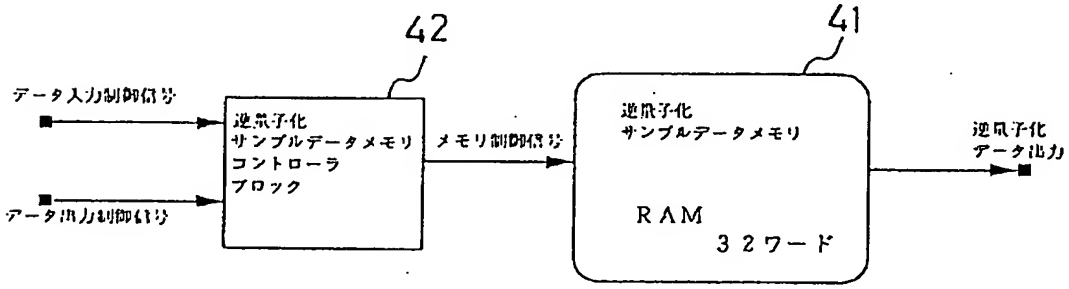
12

【図１１】本装置で用いる帯域合成処理手順を示すフローチャートである。

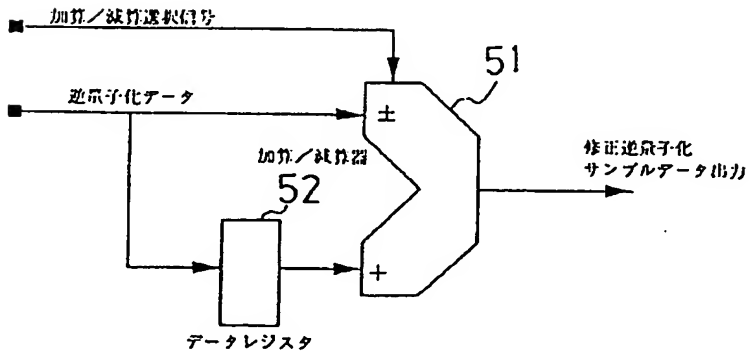
【図１】



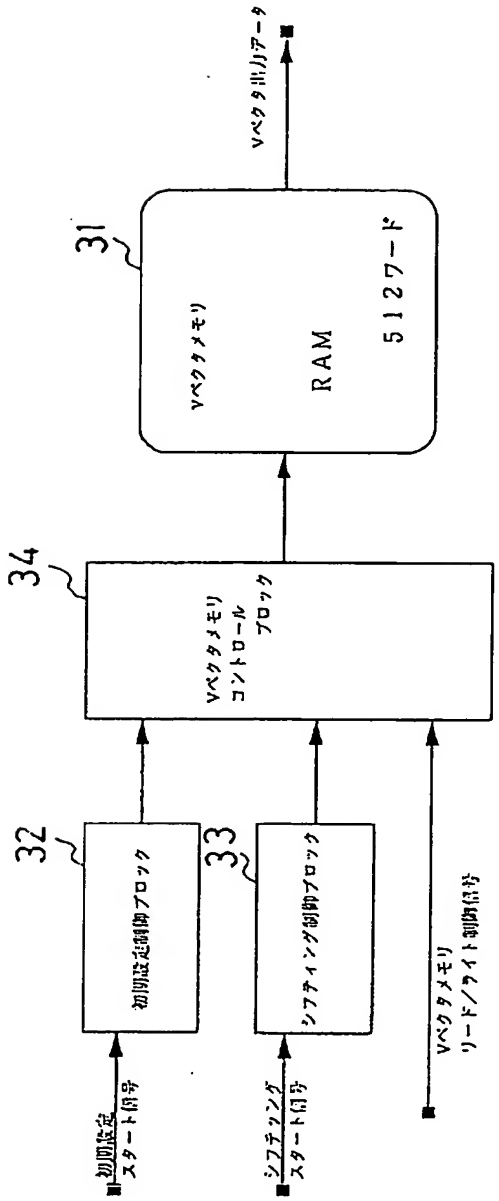
【図２】



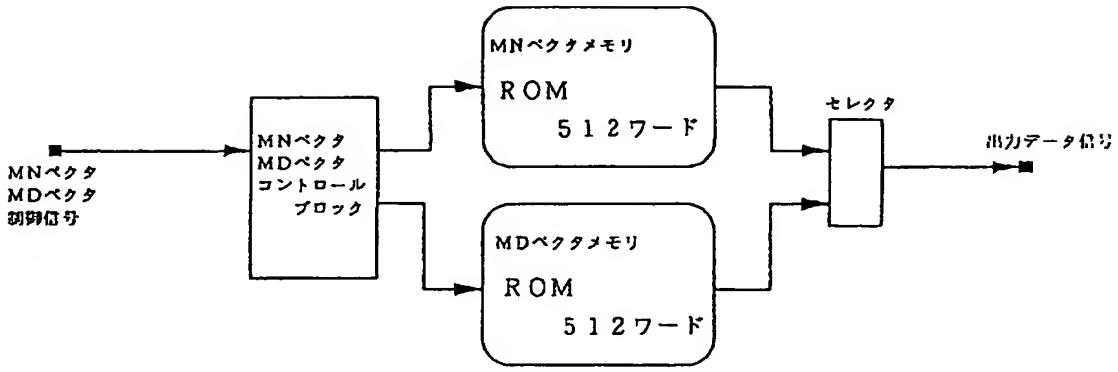
【図３】



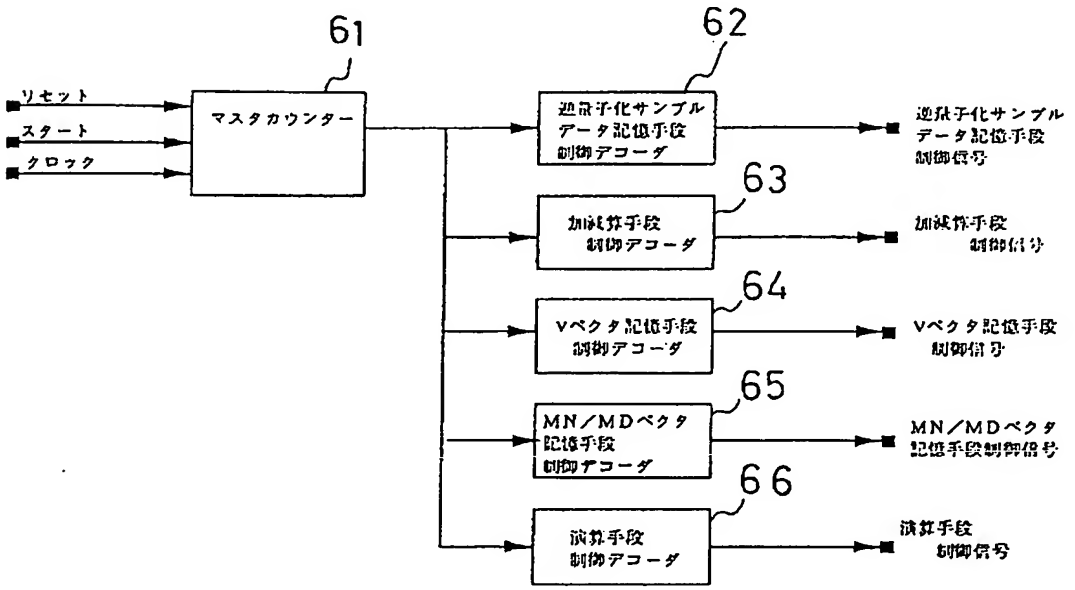
【図４】


Vベクタ出力データ
31
Vベクタメモリ
RAM
5127ード
34
Vベクタメモリ
コントロール
ブロック
32
初期設定制御ブロック
33
シフティング制御ブロック
初期設定
スタート信号
シフティング
スタート信号
Vベクタメモリ
リード/ライト制御信号

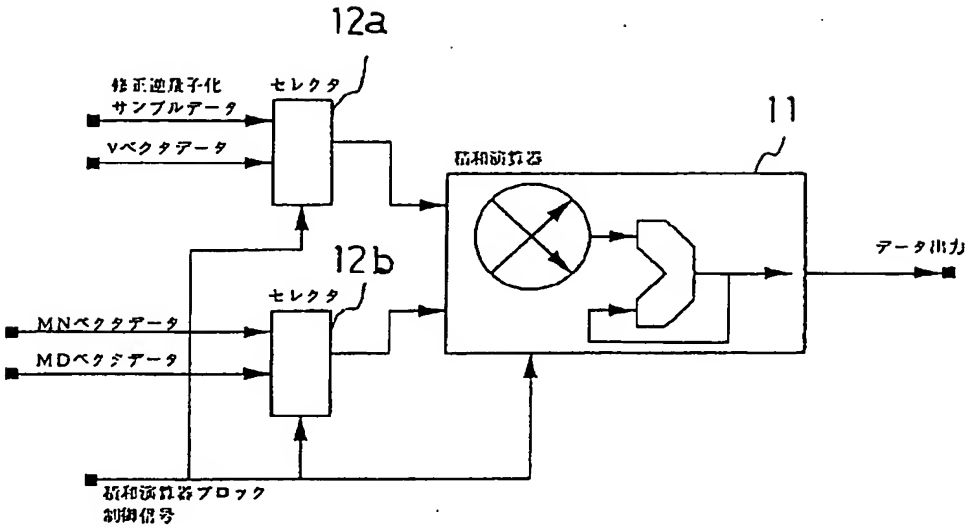
【図5】


MNベクタメモリ
ROM
512ワード
セレクタ
MNベクタ
MDベクタ
コントロール
ブロック
MNベクタ
MDベクタ
制御信号
MDベクタメモリ
ROM
512ワード

【図6】


61
62
リセット
スタート
クロック
マスタカウンター
逆量子化サンプル
データ記憶手段
制御デコーダ
逆量子化サンプル
データ記憶手段
制御信号
63
加減算手段
制御デコーダ
加減算手段
制御信号
64
Vベクタ記憶手段
制御デコーダ
Vベクタ記憶手段
制御信号
65
MN/MDベクタ
記憶手段
制御デコーダ
MN/MDベクタ
記憶手段制御信号
66
演算手段
制御デコーダ
演算手段
制御信号

【図7】


12a
修正逆量子化
サンプルデータ
セレクタ
Vベクタデータ
11
積和演算器
データ出力
12b
セレクタ
MNベクタデータ
MDベクタデータ
積和演算器ブロック
制御信号

【図８】

係数N[i][k]の性質（4）を利用して以下のように変形する

$$\begin{pmatrix} V[0] \\ V[1] \\ \vdots \\ V[2n] \\ V[2n+1] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] & N[0,15] & N[0,14] & \cdots & N[0,1] & N[0,0] \\ N[1,0] & N[1,1] & \cdots & N[1,14] & N[1,15] & -N[1,15] & -N[1,14] & \cdots & -N[1,1] & -N[1,0] \\ \vdots & & & & \vdots & \vdots & & & & \vdots \\ N[2n,0] & & \cdots & & N[2n,15] & N[2n,15] & & \cdots & & N[2n,0] \\ N[2n+1,0] & & & & N[2n+1,15] & -N[2n+1,15] & & \cdots & & -N[2n+1,0] \\ \vdots & & & & \vdots & \vdots & & & & \vdots \\ N[63,0] & N[63,1] & & N[63,14] & N[63,15] & -N[63,15] & -N[63,14] & & -N[63,1] & -N[63,0] \end{pmatrix} \begin{pmatrix} S[0] \\ S[1] \\ \vdots \\ S[2n] \\ S[2n+1] \\ \vdots \\ S[31] \end{pmatrix}$$

$$\begin{pmatrix} V[0] \\ V[1] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ N[1,0] & & & & \vdots \\ \vdots & & & & \\ N[63,0] & & \cdots & & N[63,15] \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[n] \pm S[31-n] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

偶数行は「＋」
奇数行は「－」をとる

次に係数N[i][k]の性質(1)(2)(3)を利用して以下のように変形する

$$\begin{pmatrix} V[0] \\ \vdots \\ V[15] \\ V[16] \\ V[17] \\ \vdots \\ V[32] \\ V[33] \\ \vdots \\ V[47] \\ V[48] \\ V[49] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ \vdots & & & & \\ N[15,0] & & & & \\ 0 & & & & \\ -N[15,0] & & & & \\ \vdots & & & & \\ -N[0,0] & & & & \\ N[33,0] & & & & \\ \vdots & & & & \\ N[47,0] & & & & \\ -1 & & & & \\ N[47,0] & & & & \\ \vdots & & & & \\ N[33,0] & & & & N[33,15] \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[n] \pm S[31-n] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

$$\begin{pmatrix} V[0] \\ \vdots \\ V[15] \\ V[33] \\ \vdots \\ V[47] \\ V[48] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ \vdots & & & & \vdots \\ N[15,0] & & & & \\ N[33,0] & & & & \\ \vdots & & & & \\ N[47,0] & & & & N[47,15] \\ -1 & & \cdots & & -1 \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

i = 16
　　V[i] = 0
17 <= i <= 32
　　V[i] = -V[32-i]
49 <= i <= 63
　　V[i] = V[96-i]
偶数行は「＋」
奇数行は「－」をとる

【図９】

$$j = 0 \text{ to } 31 \quad Sj = \sum_{i=0}^{15} W(j+32i)$$

$$= W(j+32*0) + W(j+32*1) + W(j+32*2) + \cdots\cdots + W(j+32*j) + \cdots\cdots + W(j+32*15)$$

$$= W(j+64*0) + W(j+64*0+32) + W(j+64*1) + W(j+64*1+32) \cdots\cdots + W(j+64*7) + W(j+64*7+32)$$

$$= \sum_{i=0}^{7} [W(j+64*i) + W(j+64*i+32)]$$

Wベクタを差し替える

$$= \sum_{i=0}^{7} [U(64*i+j) * D(64*i+j) + U(64*i+32+j) * D(64*i+32+j)]$$

さらにUベクタをV_{512}ベクタに差し替える

if (j = 0) then

$$= \sum_{i=0}^{7} [V_{512}(32*2i+j) * D(j+64*i) + -V_{512}(32*(2i+1)) * D(j+64*i+32)]$$

if (1 <= j <= 15) then

$$= \sum_{i=0}^{7} [V_{512}(32*2i+j) * D(j+64*i) + V_{512}(32*(2i+1)+j+15) * D(j+64*i+32)]$$

if (j = 16) then

$$= \sum_{i=0}^{7} [0 * D(j+64*i) + V(32*(2i+1)+j+15) * D(j+64*i+32)]$$

if (17 <= j <= 31) then

$$= \sum_{i=0}^{7} [-V_{512}(32*2i+(32-j)) * D(j+64*i) + V_{512}(32*(2i+1)+(47-j)) * D(j+64*i+32)]$$

...

各アドレスを示す関数（変数ij）を定義する

(1) F1(i,j)関数の定義

i = 2n and 0 <= j <= 16
F1(i,j) = 32*i + j

i = 2n and 17 <= j <= 31
F1(i,j) = 32*i + 32 - j

i = 2n+1 and j = 0
F1(i,j) = 32*i

i = 2n+1 and 1 <= j <= 16
F1(i,j) = 32*i + j + 15

i = 2n+1 and 17 <= j <= 31
F1(i,j) = 32*i + 47 - j

(2) F2(i,j)関数の定義

i = 2n
F1(i,j) = 64*i + j

i = 2n+1
F1(i,j) = 32*i + j + 32

(3) F3(i,j)関数の定義

i = 2n and 0<=j<=15
F3(i,j) = 1

i = 2n and j=16
F3(i,j) = 0

i = 2n and 17<=j<=31
F3(i,j) = -1

i = 2n+1 and j = 0
F3(i,j) = -1

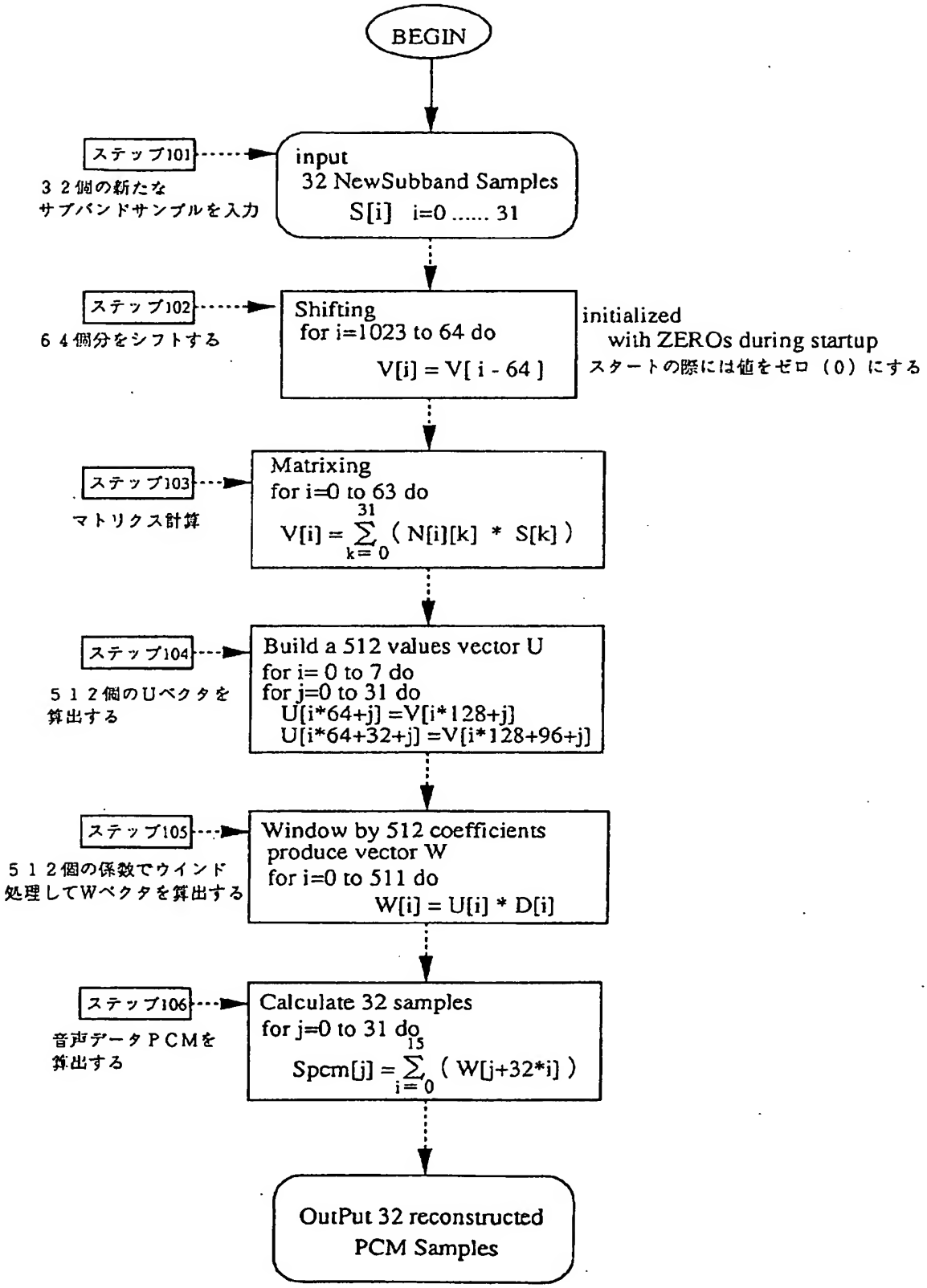
i = 2n+1 and 1<=j <= 31
F3(i,j) = 1

結果、算出式は以下のようになる

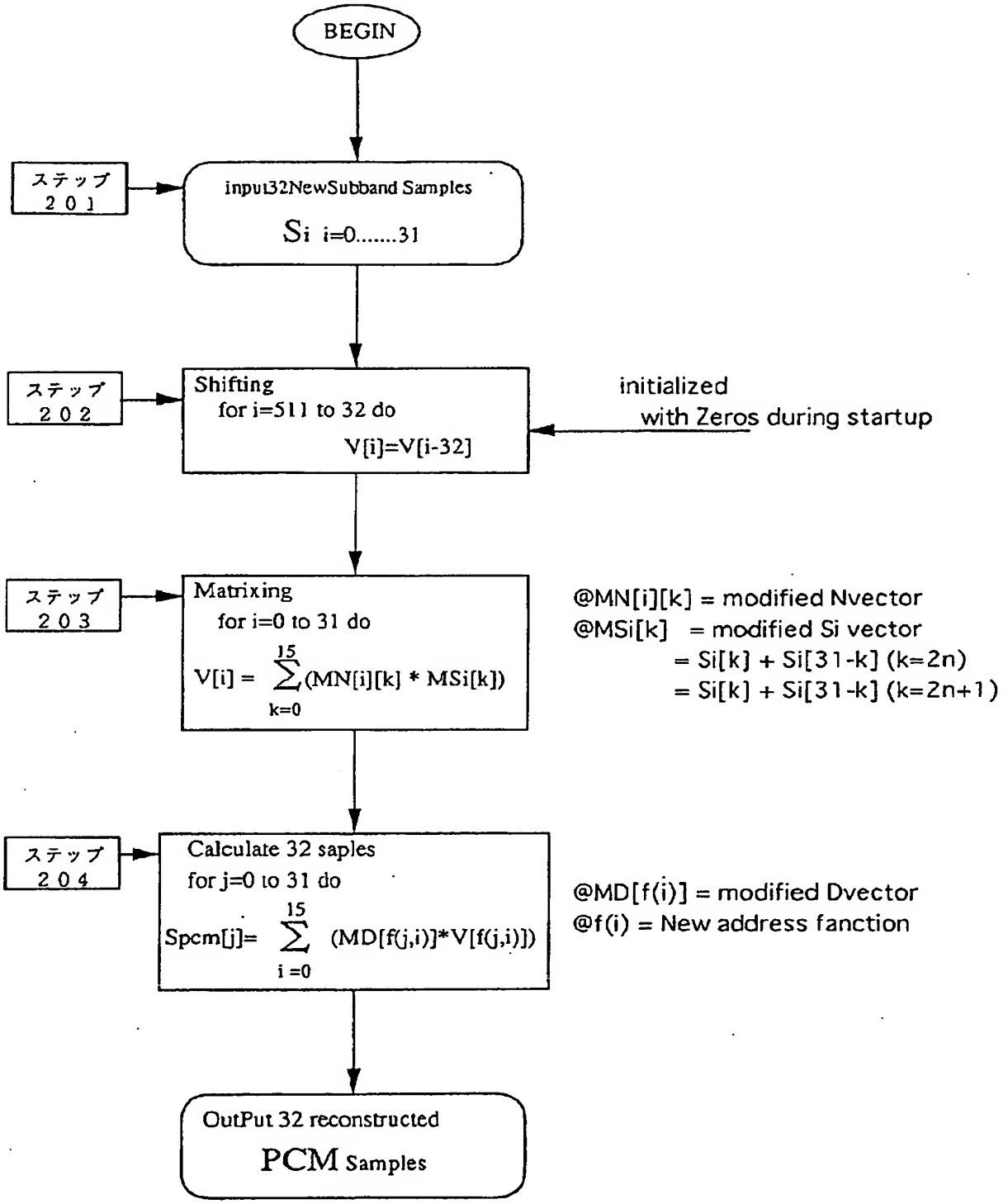
j = 0 to 31

$$S(j) = \sum_{i=0}^{15} [F3(i,j) * V512(F1(i,j)) * D(F2(i,j))]$$

【図１０】


BEGIN
ステップ101
３２個の新たな
サブバンドサンプルを入力
input
32 NewSubband Samples
S[i] i=0 31
ステップ102
６４個分をシフトする
Shifting
for i=1023 to 64 do
V[i] = V[i - 64]
initialized
with ZEROs during startup
スタートの際には値をゼロ（０）にする
ステップ103
マトリクス計算
Matrixing
for i=0 to 63 do
V[i] = Σ (k=0 to 31) (N[i][k] * S[k])
ステップ104
５１２個のUベクタを
算出する
Build a 512 values vector U
for i= 0 to 7 do
for j=0 to 31 do
U[i*64+j] =V[i*128+j]
U[i*64+32+j] =V[i*128+96+j]
ステップ105
５１２個の係数でウインド
処理してWベクタを算出する
Window by 512 coefficients
produce vector W
for i=0 to 511 do
W[i] = U[i] * D[i]
ステップ106
音声データＰＣＭを
算出する
Calculate 32 samples
for j=0 to 31 do
Spcm[j] = Σ (i=0 to 15) (W[j+32*i])
OutPut 32 reconstructed
PCM Samples

【図１１】


BEGIN
ステップ 201
input32NewSubband Samples
Si i=0.....31
ステップ 202
Shifting
for i=511 to 32 do
V[i]=V[i-32]
initialized
with Zeros during startup
ステップ 203
Matrixing
for i=0 to 31 do
V[i] = Σ (MN[i][k] * MSi[k]) k=0 to 15
@MN[i][k] = modified Nvector
@MSi[k] = modified Si vector
= Si[k] + Si[31-k] (k=2n)
= Si[k] + Si[31-k] (k=2n+1)
ステップ 204
Calculate 32 saples
for j=0 to 31 do
Spcm[j]= Σ (MD[f(j,i)]*V[f(j,i)]) i=0 to 15
@MD[f(i)] = modified Dvector
@f(i) = New address fanction
OutPut 32 reconstructed
PCM Samples